Maturing the ARM Software Ecosystem for U.S. DOE/ASC Supercomputing

SIAM PP18
March 7-10, 2018

Kevin Pedretti, Jim H. Laros III, Si Hammond
ktpedre@sandia.gov

SAND2018-2472 C
Outline

- Vanguard Petascale ARM Platform
- Tri-lab ARM Software Stack Effort
- Preliminary Results
- Conclusion
Vanguard: Prototype Systems for Advanced Architectures

- Expand the HPC ecosystem by developing emerging, yet-to-be-proven, technologies
  - Is technology viable for future production platforms supporting ASC integrated codes?
  - Increase technology choices
- Address hardware and software technologies together
  - If hardware technology is new, gaps in software stack are certain
- Buy down risk before commitment on capability/capacity class investment
Where Vanguard Fits

Test Beds
- Small testbeds (~10-100 nodes)
- Breadth of architectures Key
- Brave users

Vanguard
- Larger-scale experimental systems
- Focused efforts to mature new technologies
- Broader user-base
- NOT PRODUCTION
- Trilab resource but not formal campaign process

ATS/CTS Platforms
- Leadership-class systems (Petascale, Exascale, ...)
- Advanced technologies, sometimes first-of-kind
- Broad user-base
- PRODUCTION USE
Sandia’s NNSA/ASC ARM Platforms

- **Sept 2011**: Applied Micro X-Gene-1
  - 47 nodes

- **Cavium ThunderX1**: 
  - 32 nodes
  - Pre-GA

- **Cavium/Penguin ThunderX1**: 
  - ARM

- **Cavium/HPE/Comanche ThunderX2**
  - ARM

- **Future ASC Platforms**
  - **ARM Next Gen**: 
    - FY18 Vanguard-1 Petascale ARM Platform
  - **ARM**: 
    - Next Gen

- **TODAY**
  - **Vanguard ARM**: 
    - ARM

- **Future Platforms**
  - **Retired**
    - Applied Micro X-Gene-1
    - Cavium ThunderX1
    - Cavium/Penguin ThunderX1
  - **2015**
    - Cavium/HPE/Comanche ThunderX2
  - **2017**
    - ARM Next Gen
  - **2018**
    - Vanguard ARM
Schedule – Past and Projected

- August 2017: Formed Tri-Lab software team (Sandia, Los Alamos, Livermore, NNSA HQ)
- September 22nd – 2nd Draft RFI released
- Week of September 25th – Prime F2F presentations
- December 2017: ARM Tri-lab Software Environment (ATSE) draft
- RFP released December 20th, responses due February 8th
- January 11th – Vendor pre-proposal brief at Sandia NM CSRI
- RFP responses distributed to technical team members February 8th

- February-May 2018: Technical review, source selection, SOW development, negotiation, and contract placement
- June 2018: Initial ATSE stack release, ver. 2018.0, on early testbed
- July/August 2018: Phase 1 platform delivery begins
Construction

• Institutionally Funded
• Design/Build Contract Awarded
• All Permits Received – Site Preparation underway
• Groundbreaking Event 9/28/17
• 40% and 90% Design Reviews Completed
• 100% Design Review by 12/8/17
• Completion Data 7/15/18
• Will feature 90% liquid cooling 10% air cooling
• Thermosyphons & Air-Side Economization for Water/Energy Savings
• Solar Farm for LEED Certification
• Non-load-bearing, movable west wall for expansion (14,000 – 20,000sf)
• 7 MW power expandable to 15 MW
Outline

- Vanguard Petascale ARM Platform
- Tri-lab ARM Software Stack Effort
- Preliminary Results
- Conclusion
Vanguard Tri-Lab Software Effort

- Accelerate maturity of ARM ecosystem for ASC computing
  - Prove viability for NNSA integrated codes running at scale
  - Harden compilers, math libraries, tools, communication libraries
    - Heavily templated C++, Fortran 2003/2008, Gigabyte+ binaries, long compiles
  - Optimize performance, verify expected results

- Build integrated software stack
  - Programming env (compilers, math libs, tools, MPI, OMP, SHMEM, I/O, ...)
  - Low-level OS (optimized Linux, network, filesystems, containers/VMs, ...)
  - Job scheduling and management (WLM, app launcher, user tools, ...)
  - System management (boot, system monitoring, image management, ...)

Improve 0 to 60 time... ARM system arrival to useful work done
Acceptance Plan – Maturing the Stack

**Applications**

- Full Scale Machine Runs
  - HPCG
  - HPL
- MicroBenchmarks
  - STREAM
  - Intel MPI Benchmarks
- Compile and Run
  - VPIC (LANL)
  - NALU (SNL)
- Demonstrate
  - Vendor S/W Stack and Kernel boot
  - Lab Rebuild and Boot of Vendor S/W Stack and Kernel

**Systems Software**

- SSI Benchmarks
  - HPCG
  - HPL
- Lab/Vendor Optimization
  - SPARC (SNL)
  - ALE3D (LLNL)
  - PARTISn (LANL)
- Compile and Run
  - RAMSES (SNL)
- Demonstrate
  - Vendor S/W Stack advancements
  - Lab build and Boot of Alternative S/W Stack and Kernel

**Tools**

- Demonstrate
  - 1 Compiler Toolchain
  - 1 MPI Runtime (Full machine)

- Demonstrate
  - 2 Compiler Toolchains (min. one open source)
  - 2 MPI Runtimes (Full machine, min. one open source)

**Hardware Delivery**

- Milestone 1
  - “Acceptance” Open Network

- Milestone 2
  - Restricted Network

- Milestone 3
  - Classified Network

**Target:** 24 months

**No more than:** 12 months

**2-3 Months**
TOSS – Tri-lab Operating System Stack

- Targets commodity technology systems at NNSA Tri-labs (Lead: Livermore, Los Alamos, Sandia)
  - RHEL7 based, supports x86_64, ppc64le, and aarch64 from single source
  - ~4K packages on all archs, 200+ built for TOSS by LLNL (compilers, MPI, ...)
  - Baseline not optimized for particular system, labs optimize
  - Partnership with RHEL to add support for new hardware pre-GA

- Concerns
  - Distribution of TOSS restricted due to licensing, vendors-only for lab use
  - Not focused on user-facing programming environment
  - Missing on ARM: Nvidia cudatoolkit & driver, AMD ROCm, security scanners, backup tools, firmware tools, third party software
Cray Programming Environment (PE)

- Announced @ SC17
- Targets Cray XC50 systems, 2018H1
  - Cavium ThunderX2 64-bit ARMv8 CPUs
  - HPC tuned network stack, Aries Network
    - Cray Linux environment / SLES12-based
    - Cray programming environment
    - Cray compiling environment for ARM
  - Production-proven, same infrastructure as NNSA/ASC ATS-1 Trinity
- Concerns
  - Vendor-proprietary
  - Ability to get new software components added to Cray stack
OpenHPC

- Targets HPC Linux clusters
  - Community effort
  - Common ingredients needed to deploy and manage an HPC cluster
  - Goal to enhance modularity and interchangeability of key components
  - Current release 1.3.3, builds on Centos 7.4 or SLES12, arm64 + x86_64

- Concerns
  - Lack of integration
  - Small cluster focused, lack of hierarchy needed for scalability
  - Ability to do optimized builds
Exascale Computing Project Software, Support on ARM

- **Correctness**
  - Visualization
    - VTK-m, ALPINE, Cinema
  - Data Analysis
    - ALPINE, Cinema
  - Co-Design

- **Applications**
  - Programming Models, Development Environment, and Runtimes
    - MPI (MPICH, Open MPI), OpenMP, OpenACC, PGAS (UPC++, Global Arrays), Task-Based (ParSEC, Legion, DARMA), RAJA, Kokkos, OMPTD, Power steering
  - Math Libraries/Frameworks
    - ScalAPACK, DPLASMA, MAGMA, PETSc/TAO, Trilinos, xSDK, PEEKS, SuperLU, STRUMPACK, SUNDIALS, DTK, TASMANIAN, AMP, FleCSI, KokkosKernels, Agile Comp., DataProp, MFEM
  - Tools
    - PAPI, HPCToolkit, Darshan, Perf, portability (ROSE, Autotuning, PROTEAS), TAU, Compilers (LLVM, Flang), Mitos, MemAxes, Caliper, AID, Quo, Perf, Anal.

- **Resilience**
  - System Software, Resource Management Threading, Scheduling, Monitoring, and Control
    - Argo Global OS, Qthreads, Flux, Spindle, BEE, Spack, Sonar
  - Node OS, low-level runtimes
    - Argo OS enhancements, SNL OS project

- **Hardware interface**

14
ARM Tri-lab Software Environment (ATSE)

NNSA/ASC Application Portfolio

ATSE Programming Environment “Product” for Vanguard
Platform-optimized builds, common-look-and-feel across platforms

Containers  Native Installs  Virtual Machines

Vendor OS  TOSS  Open OS e.g. OpenSUSE

Base OS Layer

Cluster Middleware e.g. Lustre, SLURM

ATSE Packaging

User-facing Programming Env

Open Source  Limited Distribution  Closed Source  Integrator Provided  ATSE Activity

Virtual Machines

Vanguard Hardware
Integrate Components from Many Sources

- RHEL
- EPEL
- TOSS
- Vendor Software
  - Koji Build Server

- ATSE Packages
  - OpenHPC
  - Vendor Software
  - Open Build Server

- ATSE Packager

Key:
- ATSE Activity
- Open Source
- Limited Distribution
- Closed Source
- Integrator Provided
Draft ATSE Timeline for 2018

- **March**
  - Continue software stack explorations and gap analysis on testbeds
  - Setup OpenBuild server and replicate OHPC package builds for aarch64

- **April – May**
  - Develop ATSE Packager framework, ability to pull packages from TOSS, RHEL, OpenHPC OBS, vendor, and other sources
  - Identify initial component list

- **June**
  - Initial ATSE release 2018.0 on Mayer
    - Lab-distribution version: TOSS BaseOS + (ATSE-GCC | ATSE-ARM | ATSE-*
    - Open-distribution version: OpenSUSE/CentOS BaseOS + ATSE-GCC

- **July – August**
  - Linux kernel optimization and HPC patches
  - Virtual machine support

- **September – October**
  - ATSE 2018.1 release
  - Initial upstream to OpenHPC push
Outline

- Vanguard
- Tri-lab ARM Software Stack Effort
  - Preliminary Results
- Conclusion
## Compiler Dashboard

Early ThunderX2 Hardware, Single node

<table>
<thead>
<tr>
<th>Workload</th>
<th>GCC 7.1</th>
<th>Vendor A</th>
<th>Vendor B</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MiniFE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pennant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sweep3D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LULESH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenMPI 2.1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kokkos Kernels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trilinos</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NaluCFD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Things are working surprisingly well

There are some issues, but being rapidly addressed and resolved

Performance is looking good. Excellent on memory bandwidth, on par for compute. Should get significantly better with GA hardware and software tuning.

Key:
- Fastest
- Middle
- Slowest

Results from Si Hammond @ Sandia
<table>
<thead>
<tr>
<th>Workload</th>
<th>GCC 7.1</th>
<th>Vendor A</th>
<th>Vendor B</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>GUPS</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>MiniFE</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>Pennant</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>Sweep3D</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>LULESH</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>OpenMPI 2.1.2</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>Kokkos Kernels</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>Trilinos</td>
<td>Fastest</td>
<td>Middle</td>
<td>Slowest</td>
</tr>
<tr>
<td>NaluCFD</td>
<td>Slowest</td>
<td>Middle</td>
<td>Fastest</td>
</tr>
</tbody>
</table>

Key: Fastest, Middle, Slowest

Results from Si Hammond @ Sandia
Conclusion

- Vanguard allows the DOE to take necessary risks to ensure a healthy HPC ecosystem for future production mission platforms
  - Increase technology choices
  - Prove ability to run multi-physics production applications at scale
- Vanguard Tri-lab software stack effort is maturing ARM for ASC computing
  - Harden compilers, math libs, and tools
  - Optimize performance, verify expected results
  - Increase modularity and openness of software stack
  - Support traditional HPC and emerging AI + ML workloads